

# **Programmer's Manual**

**for**

# **Bake It Happen**

**Version 1.0 approved**

**Prepared by April Thompson, Daniel Bjerring, Dylan Jallo, Silvestre Flores**

**Humboldt State University**

**12/02/2020**

<b>1. Introduction</b>	<b>1</b>
<b>1.1 Purpose</b>	<b>1</b>
<b>1.2 Scope</b>	<b>1</b>
<b>1.3 Security</b>	<b>1</b>
<b>1.4 Overview</b>	<b>1</b>
<b>1.5 References</b>	<b>1</b>
<b>2. Getting Started</b>	<b>2</b>
<b>2.1 Requirements</b>	<b>2</b>
<b>2.2 Setup</b>	<b>2</b>
<b>2.3 Accessing the Application</b>	<b>2</b>
<b>2.4 File Descriptions</b>	<b>2</b>
<b>3. Coding Standards</b>	<b>3</b>
<b>3.1 HTML</b>	<b>3</b>
<b>3.2 Javascript</b>	<b>3</b>
<b>3.3 PHP</b>	<b>3</b>
<b>3.4 SQL</b>	<b>3</b>
<b>4. Modules</b>	<b>3</b>
<b>4.1 User Interface</b>	<b>4</b>
<b>4.2 Description of Modules</b>	<b>4</b>

## **Revision History**

<b>Name</b>	<b>Date</b>	<b>Reason For Changes</b>	<b>Version</b>
-------------	-------------	---------------------------	----------------


# 1. Introduction

## 1.1. Purpose

The purpose of this document is to explain the program code and programming practices used in implementing the Bake It Happen web application, with the goal of aiding developers both in debugging potential issues and in adding possible features to the web application.

## 1.2. Scope

The purpose of the project is to provide a food recipe web app that allows users to input ingredients to find recipes that consist of those exact ingredients. Users will be able to search for recipes from the Spoonacular API as well as a database of user submitted recipes. Users are able to submit their own recipes after they have logged in and they can access them in their user profile. Our intended audience for this project are parents, students and those who enjoy cooking for themselves but might not have the time or budget to make anything complicated.

## 1.3. Security

To guard against malicious intent, form sanitization is implemented with PHP by using the `strip_tags()` function. For security PHP `md5()` method of encryption is used to encrypt the user's password when saving into the database. To guard against unauthorized duplicate accounts Emails are checked to not already be in use.

## 1.4. Overview

The remainder of this document is sectioned into three succeeding components:

2. **Getting Started** – general use, set up, and logging into the application.
3. **Coding Standards** – Details usage of languages used for each feature.
4. **Modules** – describes the different functional components of the application.

## 1.5. References

Spoonacular API <https://spoonacular.com/food-api/updates> Last Updated July 2020  
Authors: Crystal (coffeebean), David (dsky), Philipp (qqilhq), Allie (almahill)

## 2. Getting Started

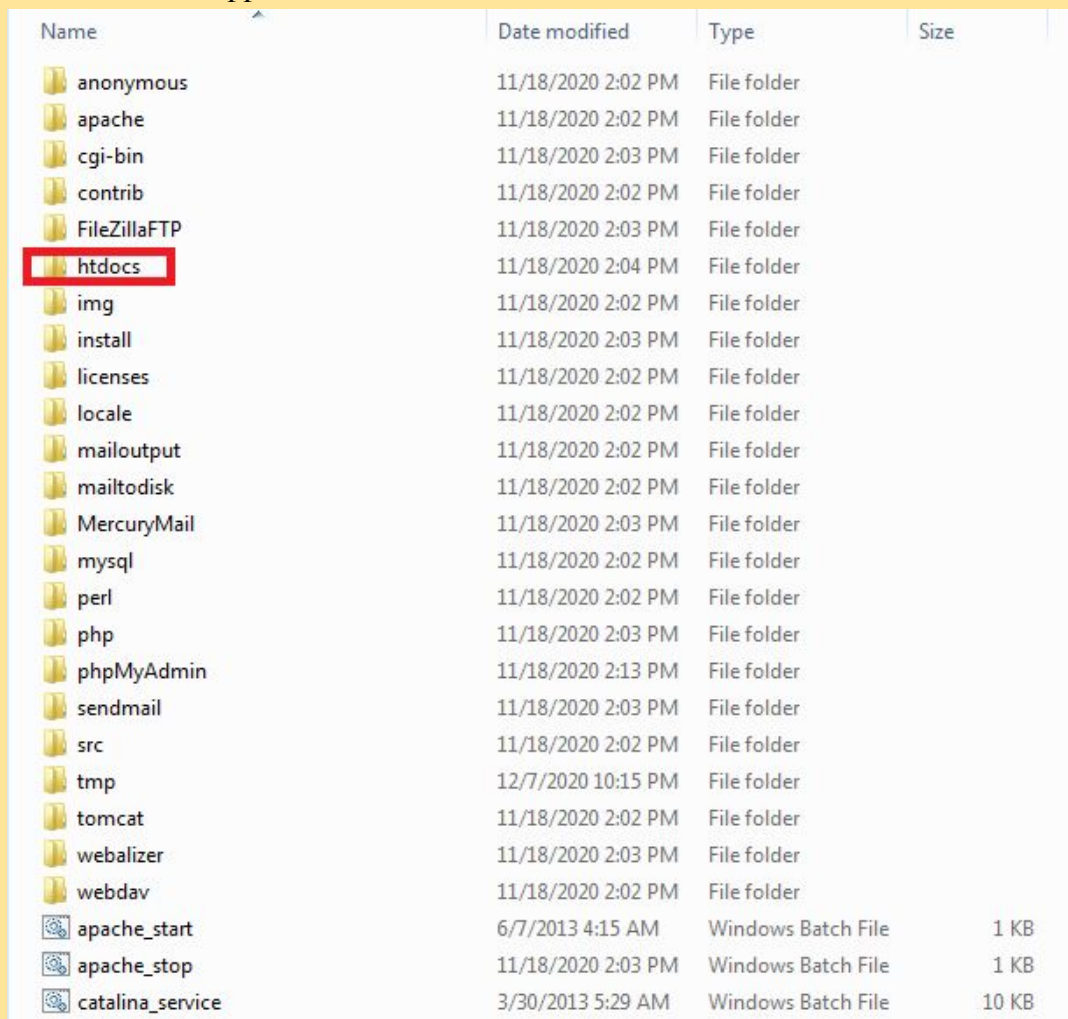
### 2.1. Requirements

To set up and run Bake It Happen, you will need:

- XAMPP 7.4.12 or later (<https://www.apachefriends.org/index.html>)
- a zip file of the current version of the repository `bakeithappenfinal` of Bake It Happen (<https://github.com/BakeItHappen/bakeithappenfinal>)
- data compression software (winZip, WinRAR, 7zip, etc.)

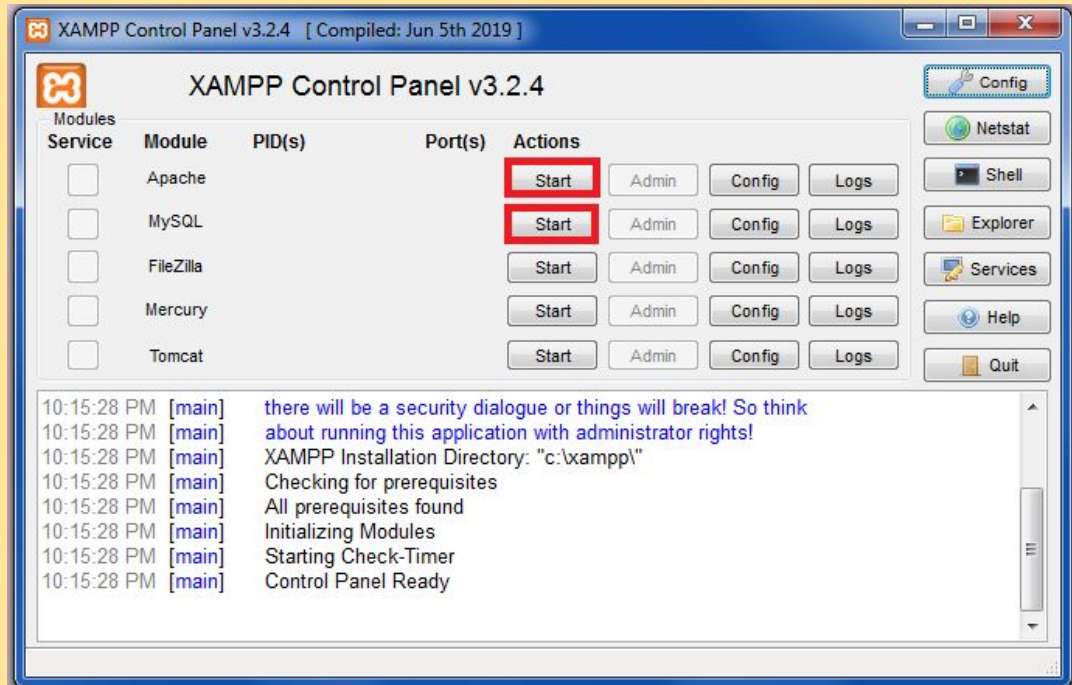
### 2.2. Setup

- In the 'htdocs' folder in your XAMPP directory create a new folder called 'Projects'. If XAMPP was installed with the default settings, the directory path should be similar to `C:\xampp\htdocs`.

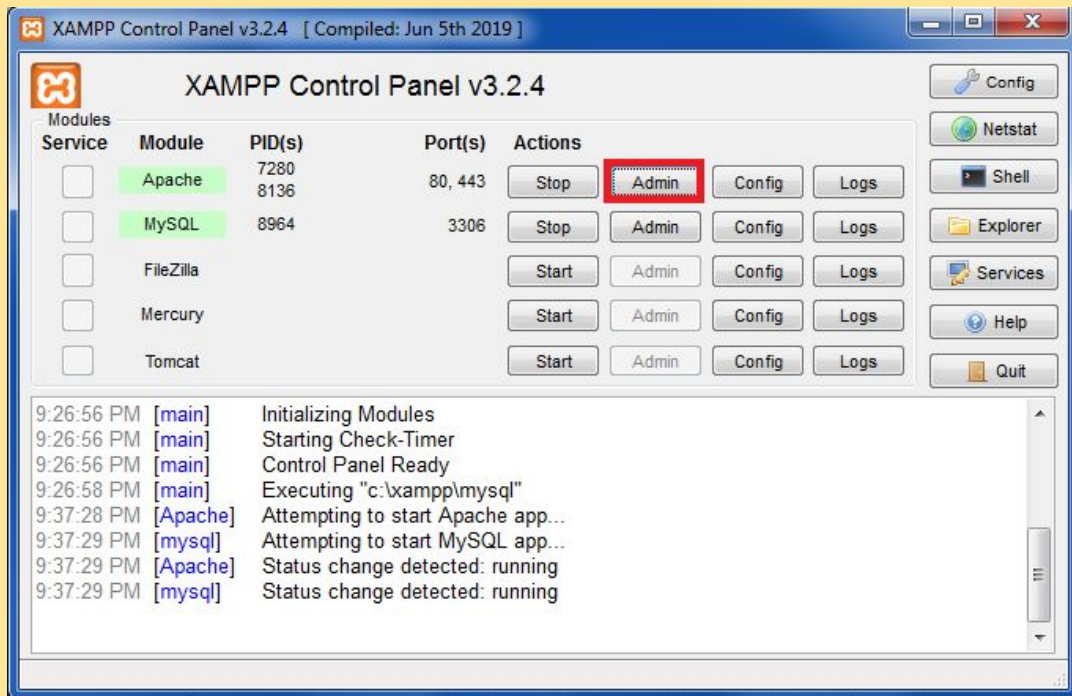


Name	Date modified	Type	Size
anonymous	11/18/2020 2:02 PM	File folder	
apache	11/18/2020 2:02 PM	File folder	
cgi-bin	11/18/2020 2:03 PM	File folder	
contrib	11/18/2020 2:02 PM	File folder	
FileZillaFTP	11/18/2020 2:03 PM	File folder	
htdocs	11/18/2020 2:04 PM	File folder	
img	11/18/2020 2:02 PM	File folder	
install	11/18/2020 2:03 PM	File folder	
licenses	11/18/2020 2:02 PM	File folder	
locale	11/18/2020 2:02 PM	File folder	
mailoutput	11/18/2020 2:02 PM	File folder	
mailtodisk	11/18/2020 2:02 PM	File folder	
MercuryMail	11/18/2020 2:03 PM	File folder	
mysql	11/18/2020 2:02 PM	File folder	
perl	11/18/2020 2:02 PM	File folder	
php	11/18/2020 2:03 PM	File folder	
phpMyAdmin	11/18/2020 2:13 PM	File folder	
sendmail	11/18/2020 2:03 PM	File folder	
src	11/18/2020 2:02 PM	File folder	
tmp	12/7/2020 10:15 PM	File folder	
tomcat	11/18/2020 2:02 PM	File folder	
webalizer	11/18/2020 2:03 PM	File folder	
webdav	11/18/2020 2:02 PM	File folder	
apache_start	6/7/2013 4:15 AM	Windows Batch File	1 KB
apache_stop	11/18/2020 2:03 PM	Windows Batch File	1 KB
catalina_service	3/30/2013 5:29 AM	Windows Batch File	10 KB

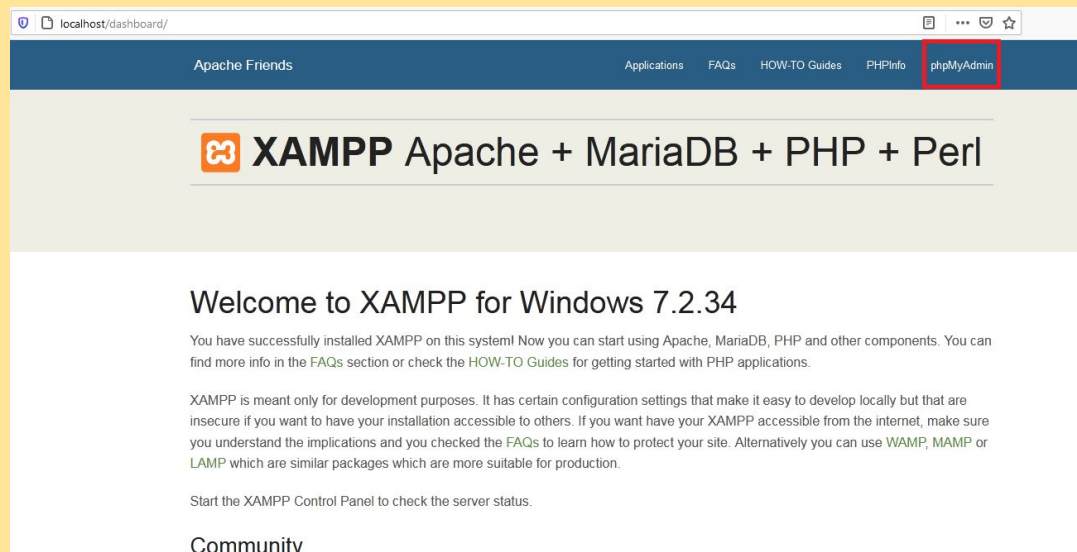
- Unzip the current version of bakeithappenfinal-master of Bake It Happen onto your computer using your preferred data compression software. Once done, move the folder named BakeItHappen into the newly created 'Projects' folder.
- File Path needs to be htdocs-> Projects -> BakeItHappen
- Once done, open the XAMPP Control Panel from your Desktop or Start Menu and click 'Start' for both the Apache and MySQL modules.



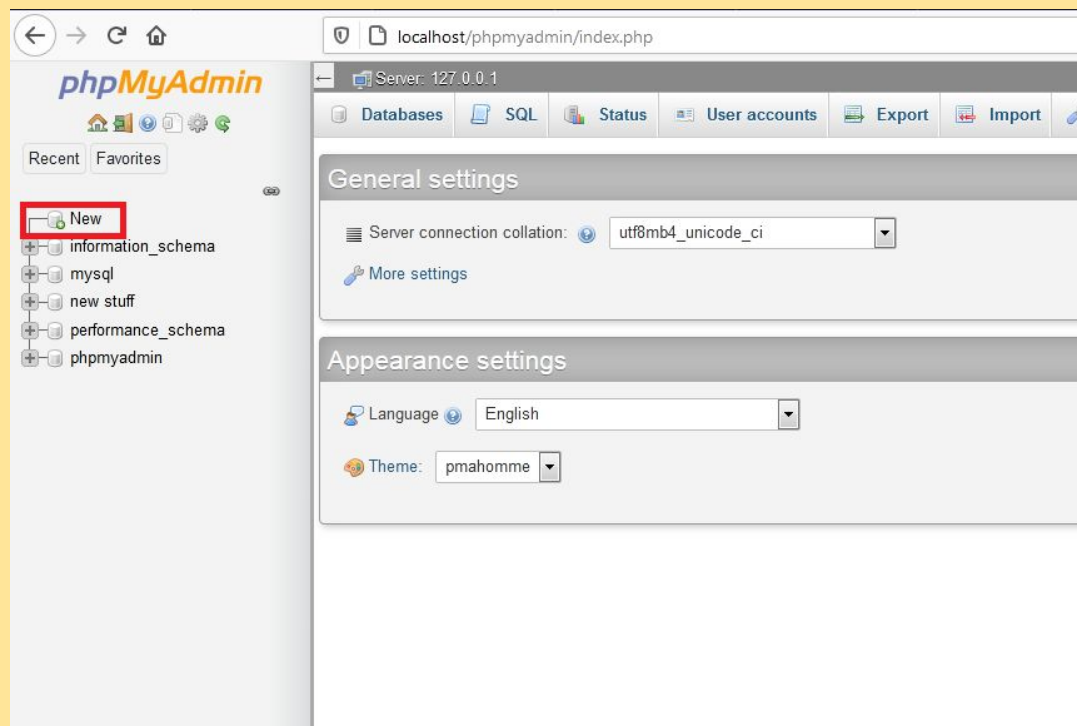
- Click 'Admin' for the Apache module.



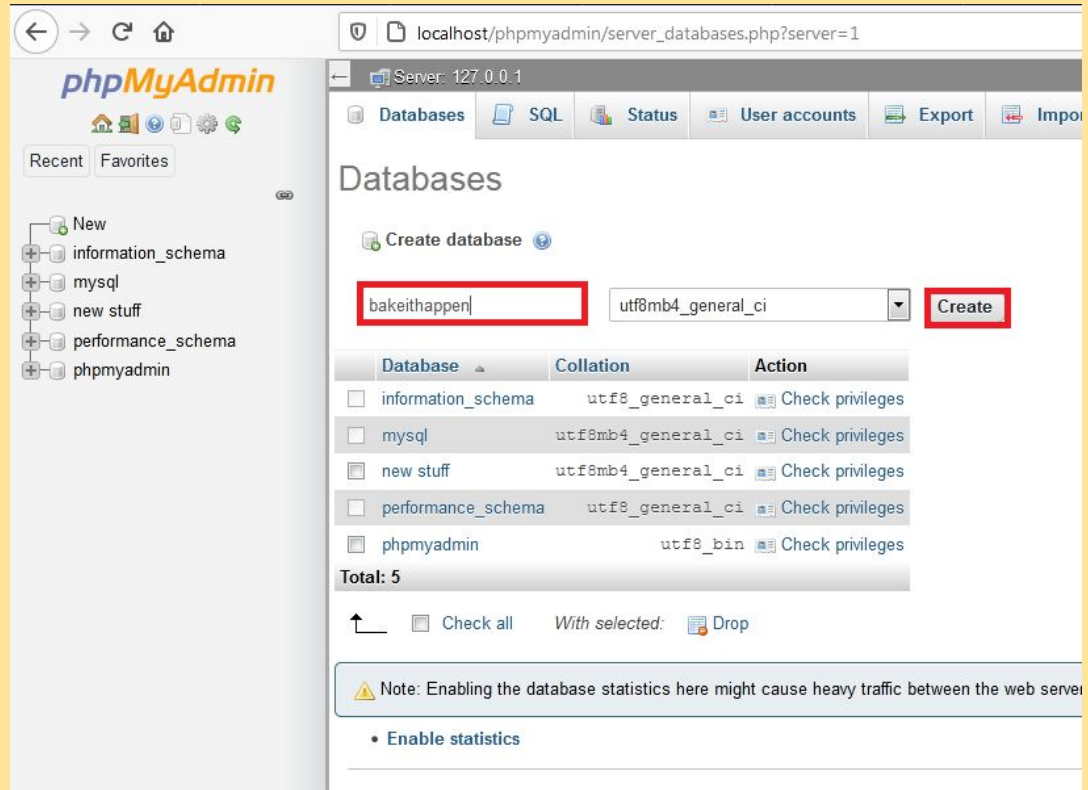
- On the <http://localhost/dashboard/> page, click 'phpMyAdmin' in the upper right corner of the screen.



- Click 'New' in the left sidebar.

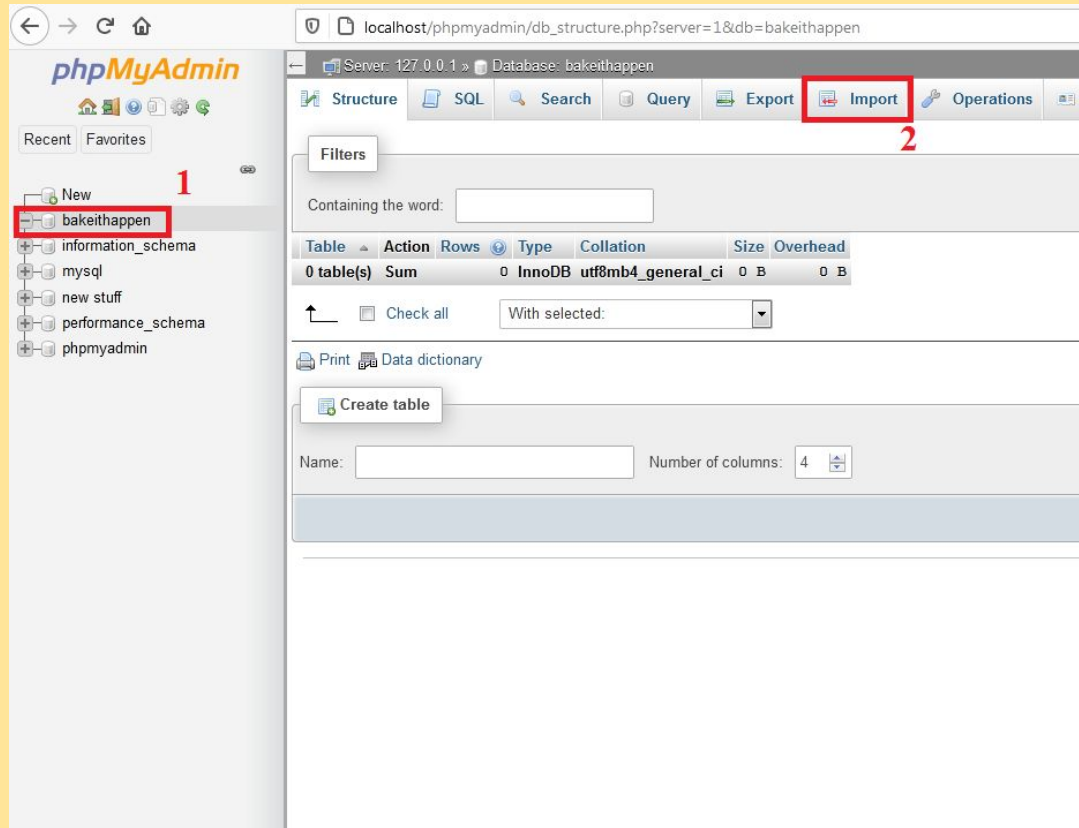


- Create a new database named 'bakeithappen' and keep all other settings unchanged.

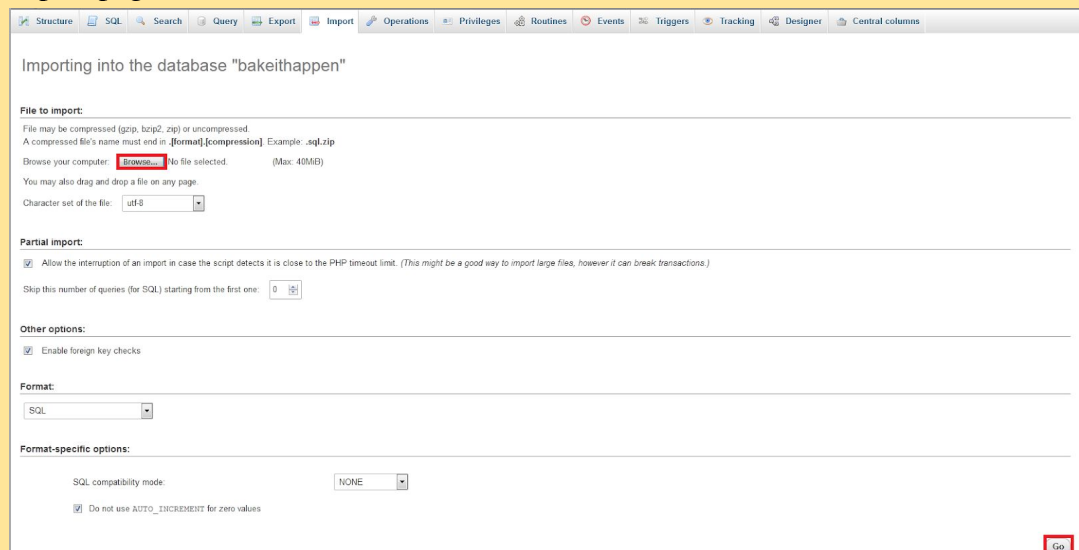


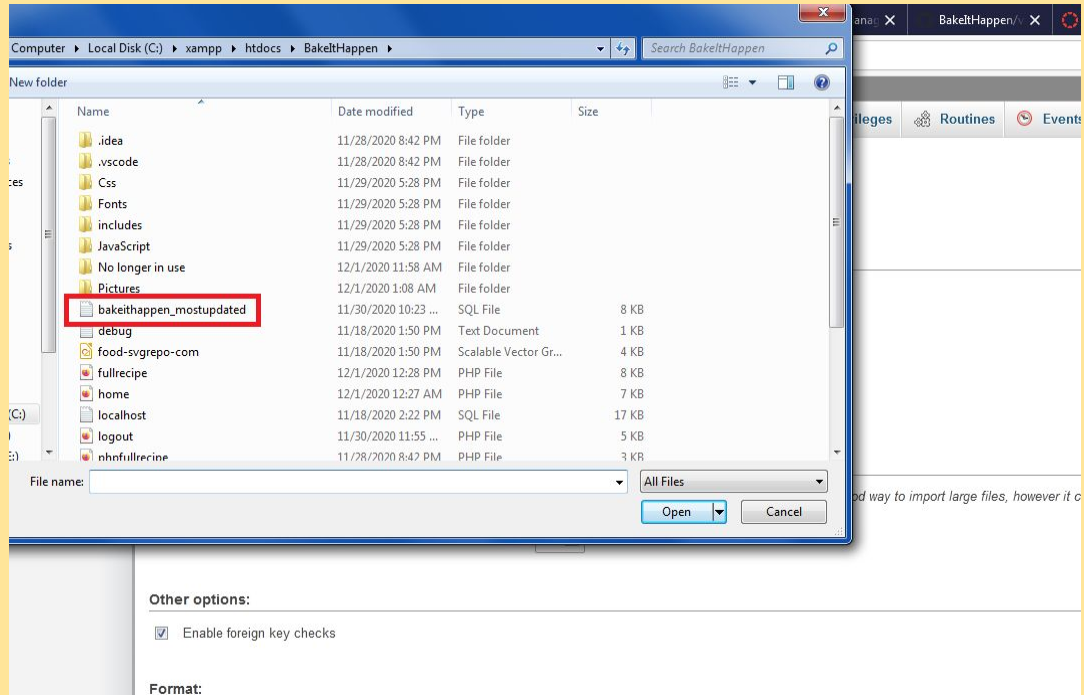
- Click the newly created 'bakeithappen' database in the left side bar, then click Import from the horizontal menu at the top of the page.



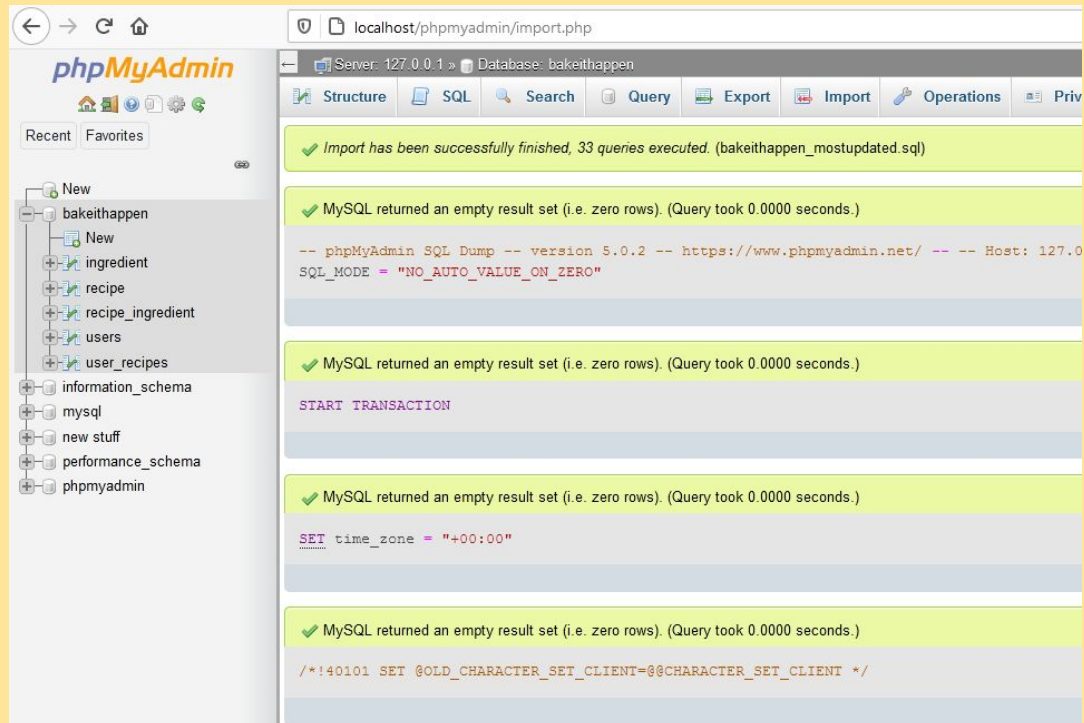


- Once there, browse to htdocs > Projects > BakeItHappen and choose 'bakeithappen\_mostupdated'. Make sure to click the 'Go' button at the bottom of the import page.





- The 'bakeithappen' happen database should now have five tables named: ingredient, recipe, recipe\_ingredient, users and user\_recipes



## 2.3. Accessing the Application

To access the Bake It Happen web application, first open the XAMPP Control Panel from your Desktop or Start Menu and click 'Start' for both the Apache and MySQL modules. Once running, open a browser of your choice and navigate to

<http://localhost/Projects/BakeItHappen/home.php>

Note: If you receive the error 'The requested URL was not found on this server' then please verify that you have the current version of BakeItHappen in the 'Projects' folder following the [set up instructions](#).

## 2.4. File Descriptions

### 2.4.1 - SQL Files

- *bakeithappen\_mostupdated.sql - Tables used for storing user data, recipe data, and used for local returns of recipes.*
- *localhost.sql - not in use old file*

### 2.4.2 - CSS Files

- *fullrecipestyle.css - CSS file, used to format fullrecipe.php*
- *mystyle.css - CSS file, used to format home.php*
- *mystyle-submit.css - CSS file, used to format submit.php, profile.php and logout.php*
- *register.css - CSS file, used to format register.php*
- *resultstyle.css - CSS file, used to format resultpage.php*

### 2.4.3 - Javascript Files

- *fullRecipe.js - Gets full recipe data from Spoonacular Api and outputs it into the html*
- *register.js - Hides login form or hides register for if text field is clicked*
- *results.js - Gets recipe results from the Spoonacular Api and outputs it into the html*
- *search.js - Gives onclick methods for the search buttons. Establishes where to go after you search.*

### 2.4.4 - PHP files

- *Account.php - Account class called by register-handler.php to validate user input and also insert user info into the database.*
- *config.php - Establishes connection to sql localhost database and starts user session.*
- *Constants.php - Constant strings for register.php error messages.*
- *fullrecipe.php - Includes HTML for displaying full recipe*

- *home.php - Landing page with search functionality. Also has links for logging in/creating accounts.*
- *login-handler.php - Starts login session and redirects to home if login button is pressed and valid credentials are entered.*
- *logout.php - Page designated for users when they click the 'Logout' link, contains a redirect back to home.php. Users should only ever see this page if something went wrong with redirection. Contains HTML that displays a successful logout message.*
- *phpfullrecipe.php - Handler for the full recipe page if the recipe is from our local results. If a "phpid" is passed through a GET method, then this code will run.*
- *profile.php - Requires users to be logged in to view. Includes HTML/PHP that displays user's information (name, user ID, email and any submitted recipes).*
- *RecInsert.php - Takes input data from submit.php, cleans the data and inserts it into the proper tables.*
- *RecSearch.php - Takes input data from the home.php search bar and returns recipes from both Spoonacular and locally stored data.*
- *register.php - Main registration form including HTML*
- *register-handler.php - Handles user input from register.php and sanitizes input before calling Account.php class for registration and redirects to home.php if registration is successful.*
- *resultpage.php - Is the returned page from RecSearch.php*
- *submit.php - Requires user to be logged in to view. Form that takes user data and sends it to RecInsert.php*

#### **2.4.5 - TrueTypeFont Files**

- *QuiteMagicalRegular-8VA2.ttf - Font used for the BakeItHappen text across all pages*

#### **2.4.6 - Image Files**

- *bakeithappentextlogo.png - Image used for logo.*
- *banner2.png - Image used for banner*
- *banner3.jpeg - Image not used*
- *banner4.png - Image used for banner rough draft*
- *bih-longheader.png - Image used for banner rough draft*
- *bihstickfigure.png - Image used for logo.*
- *board.jpeg - Image not used*
- *chopping.png - Image not used*
- *default-image.jpeg - Image not used*
- *logo.png - Image used for logo.*
- *Profile\_avatar.png - default user avatar image*
- *register.jpeg - Image not used*
- *submit\_img.png - Image used in submit page*

## 3. Coding Standards

### 3.1. HTML

- Write comments above sections and where necessary.
- Indent appropriately
- Use Bootstrap alongside custom CSS

### 3.2. Javascript

- Use Zeros instead of uppercase "O"
- Class names should be capitalized (e.g: "ClassName")
- Names should be descriptive of what the function/variable/class is for and not open to interpretation (don't use variable\_one, use ingredient\_one)
- Names should be long enough to be meaningful but short enough to avoid being wordy.
- Names should be easily readable in accordance with coding standards
- Javascript is used to output the PHP variables into the html.
- Indent appropriately

### 3.3. PHP

- If PHP is used in an HTML tag, make sure PHP tag is closed. Otherwise use PHP before or after HTML.
- Indent appropriately

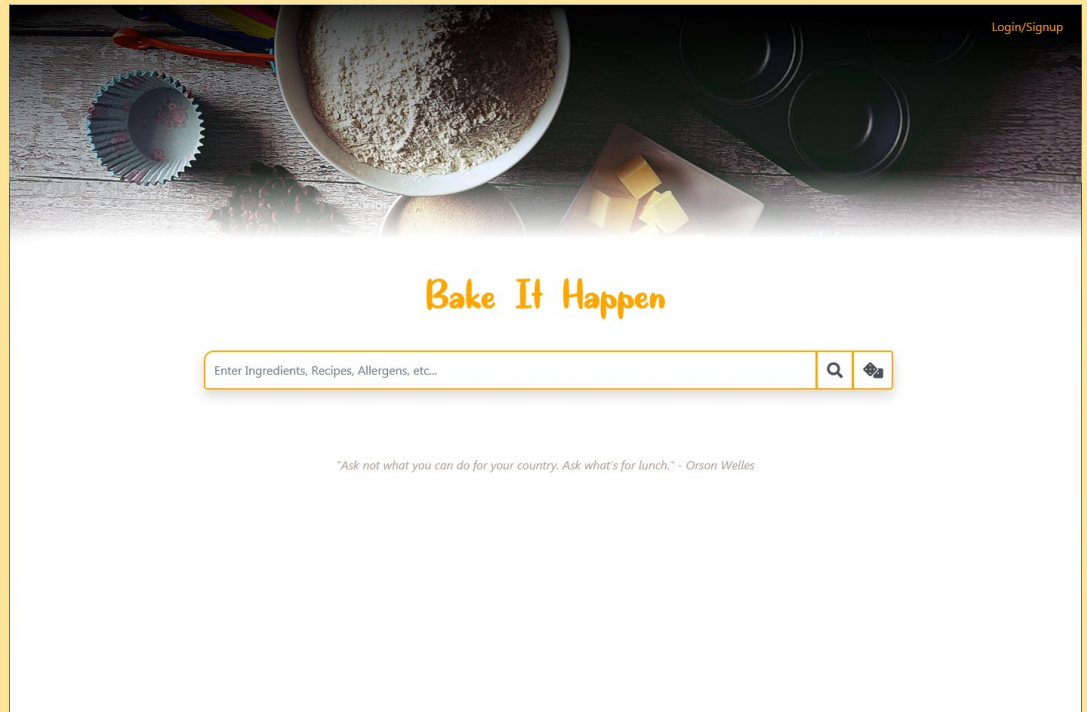
### 3.4. SQL

- Don't hardcode variables into the sql query -> use bind\_params in PHP
- Don't select \* in a query
- Always start with dropping tables.

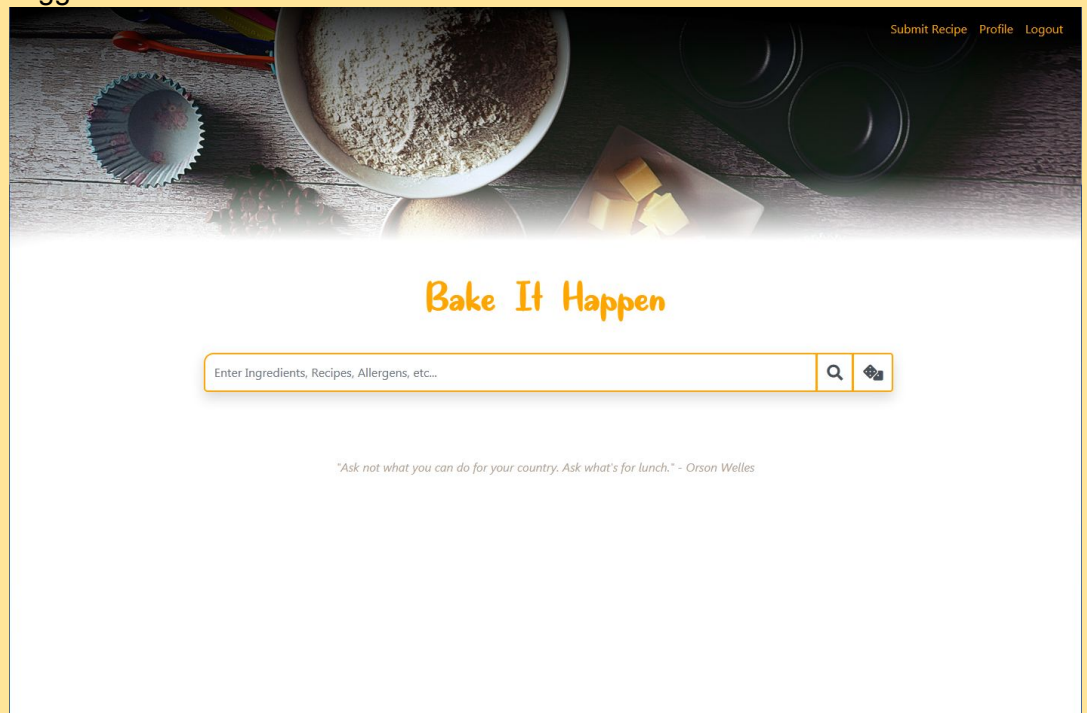
## 4. Modules

### 4.1. User Interface

- Homepage
  - Not logged in




- Logged in



- Login/Registration page
  - Login





Home

### Login to your account

Username  
e.g. bartSimpson

Password  
Your password

LOG IN

Don't have an account yet? Click here to sign up.

## ■ Registration



### Create your free account

Username  
e.g. bartSimpson

First name  
e.g. Bart,

Last name  
e.g. Simpson

Select image to upload:  No file selected.

Email  
e.g. bart@gmail.com

Confirm email  
e.g. bart@gmail.com

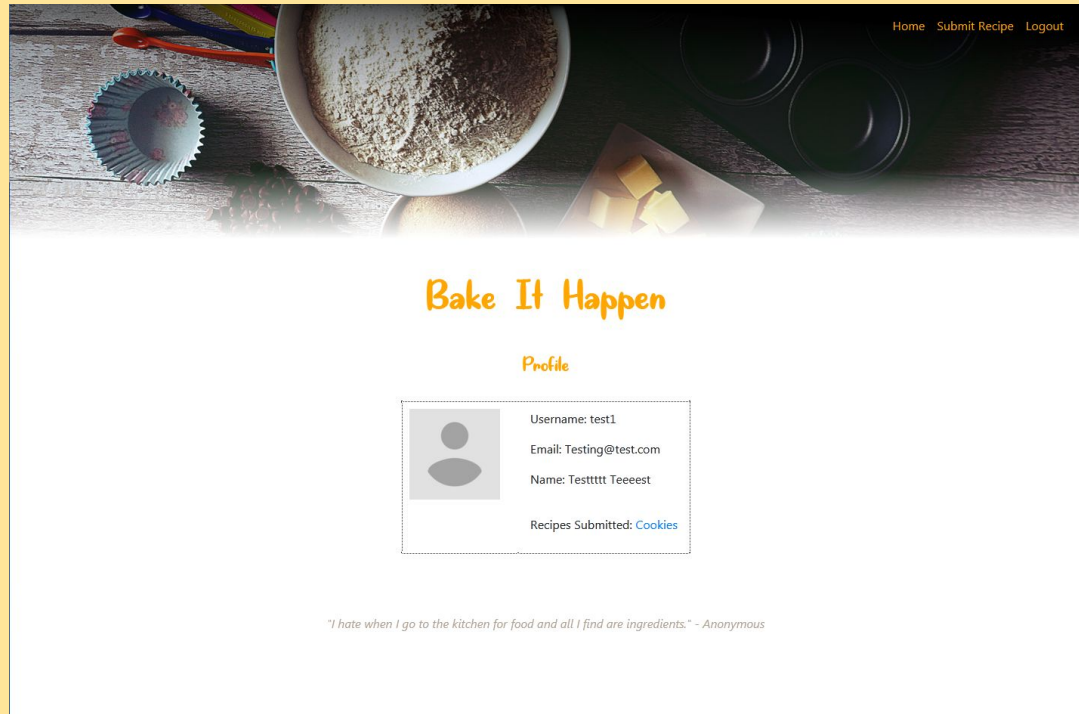
Password  
Your password

Confirm password  
Your password

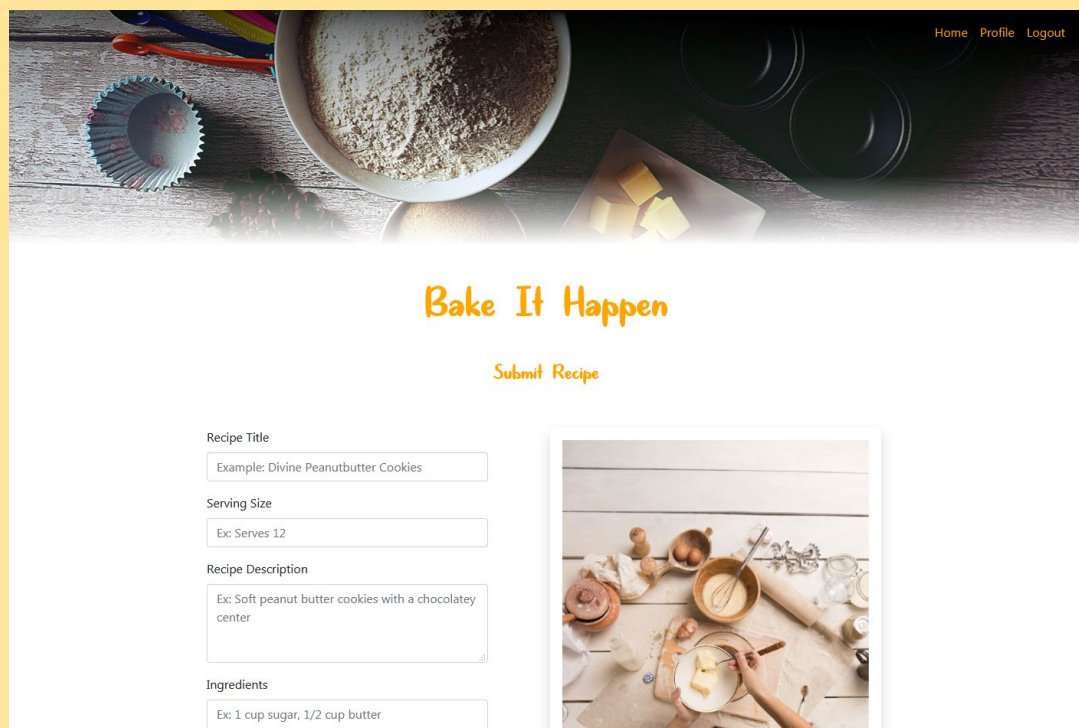
SIGN UP

Already have an account? Click here to login.

- Profile Page



- Recipe Submit page





**Submit Recipe**

Recipe Title


Serving Size

Recipe Description

Ingredients


Directions

[Submit](#)




*"I hate when I go to the kitchen for food and all I find are ingredients." - Anonymous*



- Recipe Search results



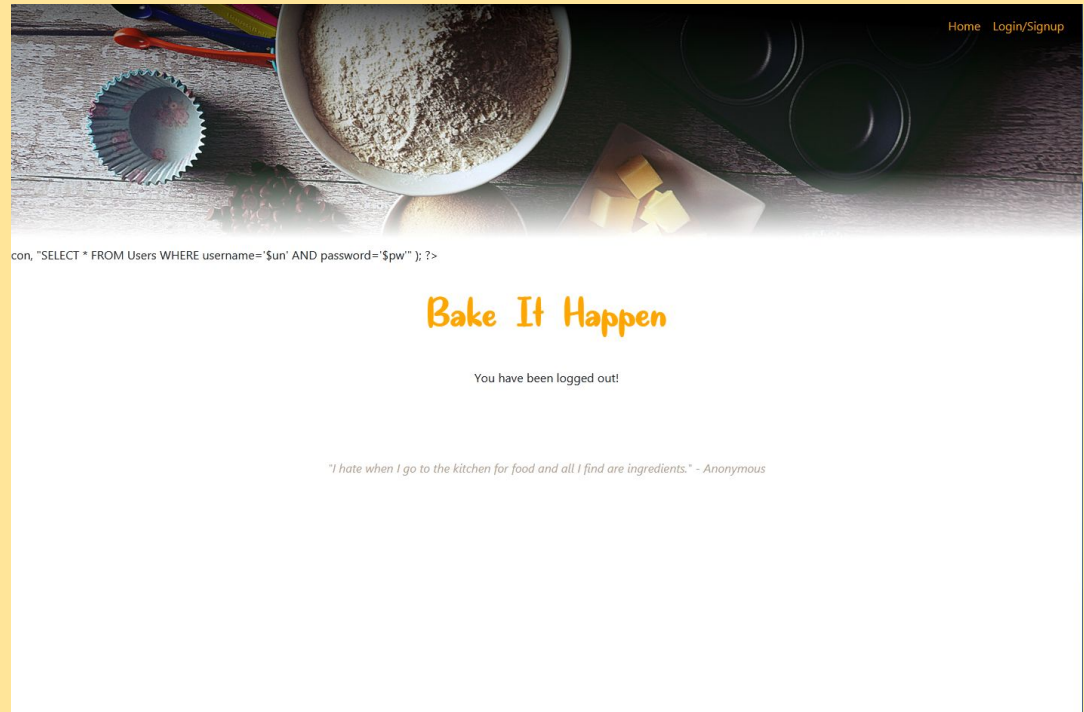
[Submit Recipe](#)
[Profile](#)
[Logout](#)



## fish

	Fish Dog
	Fish Pie

- Logout page
  - Should not be seen - if seen then redirect to home.php is broken



## 4.2. Description of Modules

- **Search Capabilities**

- **Spoonacular API**

- Our search function in the search.js, results.js, and fullrecipe.js access the Spoonacular API to get a robust amount of search results
    - We are using the free tier(only 150 points allowed a day)
    - Each request to get a recipe takes a point
    - An api key is necessary and is found on the top of the javascript pages as a variable
    - Each request made to the API needs to have the api key and pass a recipe id or a request in the url, with the format for example:

```
○ fullRecipeUrl = "https://api.spoonacular.com/recipes/"+  
  recipeIdFromStorage +"/analyzedInstructions?apiKey=" +  
  APIKEY;
```

- Data is passed asynchronously to and from the api
    - Link to Spoonacular API: <https://spoonacular.com/food-api>

- **Standard Recipe Search**

- Inputs must be separated by a space. (blank space used as delimiter)

- A standard search can contain any number of inputs. (By inputting nothing a randomized search is triggered)
- Returned results will be sorted by local first.
- Results come from both spoonacular api and our own database

### **- Randomized Recipe Search**

- Can be accessed through hitting the Standard Search button, with no inputs, resulting in many randomized results.
- Can also be accessed through hitting the “Dice” button returning a single randomized result.

## **• Registration/Login**

### **- Registration**

- If the length of the username is greater than 25 or less than 5, an error message is inserted into the error array.
- If the length of the first or last name is greater than 25 or less than 2, an error message is inserted into the error array.
- To check if a username already exists, an sql query is performed and if the username being created is already in the database, an error message is inserted into the error array.
- To validate the email, the user enters it twice and the two strings are compared, if they are not equal, an error message is inserted into the error array.
- To ensure a valid email address is entered the PHP filter\_var() function is used with FILTER\_VALIDATE\_EMAIL.
- To check if an email is not already in use, an sql query is performed and if the email being created is already in the database, an error message is inserted into the error array.
- To validate the password, the user enters it twice and the two strings are compared, if they are not equal, an error message is inserted into the error array.
- If the length of the password is greater than 25 or less than 5, an error message is inserted into the error array.
- preg\_match() function and regex is used to ensure the password includes only alphanumeric characters if not, an error message is inserted into the error array.

### **-Logging In**

- After creating an account, users may log into a personal account.
- First field for the username must be entered correctly.
- Second field for the password must be entered correctly.
- Once the login button is clicked, login-handler.php calls Account.php login function, and an sql query is performed to validate credentials. If credentials are valid the user is redirected to home.php.

- **User Submissions**

- **User Submitted Recipes**

- Users must be logged in to submit recipes.
    - Recipe must contain at least 1 ingredient.
    - Ingredients must be separated by a space.

- **Accessing User Submitted Recipes**

- User submitted recipes are accessed through the same process that accesses Spoonacular recipes.